COLUMN

シリコンバレーに学ぶキャリア・デベロップメント

第14回

利殖とキャリア開発



東原朋成

ITバブルのころからの風潮なのでしょうか．若手エンジニアの中には有望なスタートアップ企業，または上場して堅実にビジネスをやっているが，現在のストック・オプション価格が比較的低い企業に入り，30歳代前半までにいつでも引退できる資産を作りたいと思っている人の割合がかなり高いと聞きます．たしかにITバブルのころには，20歳代の億万長者をずいぶん生み出しました（今はGoogle長者を除けばまれ）．だれもがそういったチャンスにかけたわけです．

また，当時は人手不足が賃金をうなぎのぼりに上昇させました．HTMLでWebページが作れる程度のエンジニアでも13万ドル以上の年収があったといいますから，たいへんなものです．こういった層が高級車を購入し，高級アパートで暮らし始め，さらには家を購入しました．しかし現在，HTMLでWebページが作れる程度では，どんなにがんばっても7万ドルも稼げません．いったん高級な生活に慣れてしまうと，かつての生活にはなかなか戻れません．未払いのローンが残っていると，2～3のしごとを掛け持ちしなければならず，かなりたいへんなことになっているケースもあると聞きます．若いうちに，一時的に高給取りになってしまった世代の苦労と言えるのかもしれません．

かのアインシュタインによれば，人類最大の発明（発見？）は「compound interest（複利）」だそうです．これは，早いうちに貯金を始めた人は，遅くなってから倍額，3倍額貯金をする人より圧倒的に多くの資産を作れるということを意味します．後になって余裕ができたら始めようとしても，先に始めた人にはなかなか追いつけないということです（ITバブルのようなことがあれば，事情が変わってくるが…）．

この概念は，預貯金だけに限ったものではありません．しごとも後になってからがんばろう，後になってからチャレンジしようといったスタンス（マインド・セット）では，先に始めた人になかなか追いつくことができません．

もちろん，例えば非常に安全だけれど0.1％とほとんど利子の付かない預貯金をしていたとすると，後から始めた人でも追いつける可能性が大きくなります．先に始めたといっても，“それなり”に始めただけでは人には追いつかれてしまいます．さらなるキャリア・アップを目ざすのであれば，それなりに始めるということはないはずです．

筆者には，小さな会社なら社長級，大会社なら部門長級のポジション以上の実力（実績）を持つ友人がいます．年齢的にはわずかしか離れていないのですが，（アセンブリ言語の）ソフトウェア・エンジニアとしてしごとを始め，将来は自分の会社を持つということを，大学卒業までに決めていたそうです．そして，実際にソフトウェア・エンジニアとして5年ほど開発に従事した後，本格的に会社設立に至る過程のシミュレーションを繰り返し，その目標に向けてしごとを変えていきました．カスタマ・サポートやマーケティング部門を経験し，その間に大学のコース（学位ではないが終了証が出るコース）をいくつか修め，足元を固めていったそうです．また，経験を積む過程で，将来いっしょにやっていける（いきたい）同僚の目星までつけていったと言います．

これだけ用意周到に物事を運んでいくことは，決して例外的なことではありません．彼の仲間の集まりに招待されたことがあります（日本についての話題がその日のテーマだった）．“類は友を呼ぶ”とはこのことだなと思わせるほど，彼と似たような経歴の持ち主がたくさん集まっていました．だれもが持っている基本姿勢は「先に始める，先に始めても投資効果の悪いことはしない」ということでした．利殖の基本姿勢を貫いているわけです．利殖は時間と利率（ゲイン），そしてリスクがもっとも重要な要素です．これら三つの要素に尽きるとも言えそうです．

ある程度たまってしまうと，もうなくならない（利子が生み出すものが消費する量より多くなるので，使っても使っても増え続ける）ことをお金持ちは知っています．それを知っている人がお金持ちになるのです．このアナロジは，そのままキャリア形成にも当てはまるということのようなのです．宝くじに当たることなどで“にわか成金”が生まれるように，運がキャリアをもたらすこともありますが，彼/彼女らは運頼みをしないわけです．

現在日本に多くいるというニートは，時間，利率，そしてリスクのどの要素も筆者の友人（とそのサークル）の対極にあるように見えます．宝くじにでも当たればよいのですが，当たらない確率のほうが高いわけですから，非常に不利だということになります．

キャリア形成と利殖には，共通点が多そうです．そういえば利殖にたけた国民性のエンジニア（マネージャ）がシリコンバレーで幅をきかせているような感じがしないでもありません．エンジニアといえども利殖について学んでおくことは，キャリア

形成や生涯計画において重要な要素なのかもしれません．

筆者が知人から聞いたエンジニアの話です．子供が12歳くらいになったときに，1,000ドル程度を渡したそうです．おこづかいではありません．投資の基本を教えて運用させるためです．この1,000ドルは，その子供が18歳になったころには高級車を購入するのに十分な金額まで殖えていたと言いますから，たいしたものです．こうやって利殖を肌で経験させ，利殖，ひいてはキャリア形成や生涯計画まで教えているのかもしれません．

インタビュー模擬体験

答 前回の解答（2006年3月号，pp.120-121）

問題

次の記述について，おかしな点を指摘してください．

(1)
```
#include <stdio.h>
#include <stdlib.h>
int main()
{
  char name[50];
  FILE *infile;
  printf("Type the filename: ");
  fgets(name, sizeof(name), stdin);
  infile = fopen(name, "r");
  if (infile == NULL) {
    printf("ERROR¥n");
    exit(1);
  }
  fclose(infile);
}
```

(2)
```
int table[30][40];
#define init_table()
        memset(table, 1, sizeof(table));
```

(3)
```
#perl
  $x='me';
  $y='you';
  print "is x and y equal?",
              $x==$y, "¥n";
```

すべて基本的な問題ばかりでした．しかし，思い込みからデバッグを必要以上に手間取らせるようなケースです．

(1) ¥nを取り除くことを忘れていることが問題になります．fgets関数で文字列を読み込むとき，末尾（¥0の直前）に¥nが入ってしまいますが，これを取り除く必要があります．

C言語を勉強するすべての人が手にするであろう著名な「プログラミング言語C ANSI規格準拠」（B.W. カーニハン，D.M. リッチー 著，石田晴久訳，共立出版）にはふんだんに出てきますから，一度は出会ったことがあるのではないでしょうか．

(2) memsetは，バイトを単位にしています（man memsetで確認）．ところが，32ビット・プロセッサであればint型は4バイトで構成されます．memsetで1バイトずつデータをセットしたとすると，0x01010101となってしまいます．tableに1をセットしようとした試みは失敗に終わってしまうわけです．

筆者は今でもライブラリ・ルーチンを誤って使用してしまうことがあります．manコマンドなどでちょっと確認すればよいのに，これを怠ったがためにデバッグで苦労する場合があります．

(3) この場合，==で比較することができません．==をeqと置き換える必要があります．

Perlのいやらしいところは，データ・タイプをどうにでもできてしまうことです．もっとも，これだけにとどまらず，困ったところがいくつもあります．

Perlを使えることが重要になってきているのですが，最近，簡単なことにはWindowsのツール（Excelなど）を使う機会が増えています．特別なスクリプトを書く必要がないため，これまでUNIX（Linux）環境で開発してきたエンジニアには，新鮮なものがあります．

問 今月の問題

問題

マネージメントの観点から，「ストラテジ」について説明してください．

ストラテジを立てることは重要です．ところでこのストラテジにはいろいろな概念があります．とくに概念の点から，その変遷などを調べてみましょう．概念の基本を知ることは，ストラテジがいかなるものかを知る手がかりになります．意外と自分が思っていることと違っているかもしれません．

この問題には，さまざまな解答が考えられると思います．次号では，その一例を紹介します．

とうはら・ともなり